Outlier Detection for High-dimensional Data Streams

Ji Zhang[1], Qigang Gao[1], Hai Wang[2]
[1]Faculty of Computer Science, Dalhousie University
{jiz,qggao}@cs.dal.ca
[2] Sobey School of Business, Saint Mary's University
hwang@smu.ca

Abstract

The explosion of data streams has sparked a lot of research interests in data mining on streaming data flow in recent years. Many data streams are inherently high dimensional and outlier detection from these data streams can potentially lead to discovery of useful abnormal and irregular patterns hidden in the streams. Outlier detection in data streams can be useful in many fields such as analysis and monitoring of network traffic data, web log, sensor networks and financial transactions.

1 Introduction

A data stream is a real-time, continuous, ordered (implicitly by arrival time or explicitly by timestamp) sequence of items. Outlier detection from stream data aims to find abnormal items (objects or points) that are abnormal or irregular with respect to majority of items in the stream. This problem is not trivial to solve whatsoever due to the following reasons. First, outlier detection for high-dimension data alone is difficult to deal with. Outliers may be embedded in some subspaces (subset of features) of full data space and the exhaustive search for the outlying subspaces is a NP problem. Secondly, the nature of streaming data further complicates this problem. It is required that data stream mining algorithms take only one pass over the data stream and process data on an incremental and real-time paradigm, subjecting to a main memory constraint. There are few research efforts on outlier detection in high-dimensional data streams so far and the existing outlier detection methods such as [1] and outlying subspace detection method such as [3] for high-dimensional data have their limitations in dealing with this problem.

2 Problem Formulation

The outlier detection problem for high-dimensional data streams can be formulated as follows: given a data stream $\mathcal{D}$ with N φ-dimensional data points, each data point $X_i = \{x_{i1}, x_{i2}, \ldots, x_{i\varphi}\}$ $(1 \leq i \leq N)$ in $\mathcal{D}$ will be classified as either an outlier or a regular data point and, if X_i is an outlier, the associated subspace(s) in which X_i is considered as an outlier will be given as well. More precisely, the outlier detection algorithm functions as a classification function $f : X_i \rightarrow < b, \mathcal{S}_i >$, where b is a boolean variable ('true' or 'false') indicating whether X_i is an outlier or not and $\mathcal{S}_i$ is the set of outlying subspaces of X_i. In the event that X_i is not an outlier, $\mathcal{S}_i$ is simple an empty set.

3 Methodology

Like many other data mining tasks on data stream analysis such as clustering and frequent item detection, one of central problems involved here is the design of appropriate statistics

over data streams that are suitable for outlier detection purpose. In our work, we employ *Evolving Cell Structure* (ECS), a new structure consisting of key statistics that are able to capture the major underlying characteristics of the data stream for outlier detection. Quantitization of ECS entails a data space partition. The data space will be partitioned into equal volumed cells and an ECS is used for each populated cell to capture the data statistics in that cell. Note that, even though the number of cells may increase dramatically when the dimension of data goes up, the number of populated cells is actually of a manageable size [2].

The ECS of a cell c in the hypercube is defined as $ECS(c) = \{n_c, \mu_c, \sigma_c\}$, where n_c is the number of points in c, μ_c is the centriod of points in c and σ_c is the standard deviation of points in c. As pointed out in [1], the points falling to a cell in the hypercube can be assumed to fit normal distribution if the granularity of the partition is sufficiently small. Hence, a given data point X_i in $\mathcal{D}$ is considered outlying in a cell c if $dist(X_i, \mu_c) \geq \alpha\sigma_c$. Intuitively speaking, a point is detected as an outlier in a cell if its distance to the centriod is a few times (stipulated by α) larger than the standard deviation. An outlying subspace of X_i is then a subspace in which X_i is an outlier in a particular cell of this subspace.

As streaming data arrive continuously, ECS thus needs to be updated dynamically. Notice that the ECS at the moment when X_i in $\mathcal{D}$ is being processed is the statistics quantitized over the past $(i-1)^{th}$ points that have been seen thus far. Processing each point in the stream involves two major tasks: (i) the point is checked in different subspaces to see whether or not it is an outlier; and (ii) ECS of the cells in different subspaces to which the point belongs are updated.

Since the number of subspaces grows exponentially with regard to dimensions of the data set, and evaluating each data point in each possible subspace is prohibitively expensive, we thus only check each point in a few subspaces in the space lattice alternatively. These subspaces can be obtained by a training process using a batch of data points in the stream. Genetic algorithm is employed to find a specific number of top subspaces which are able to detect as many outliers as possible from the training data. These top subspaces will be updated periodically over time to allow for possible concept drift in the data stream.

4 Conclusions and Future Work

In this paper, we tackled the outlier detection problem on high-dimensional data streams. A new statistical structure, called Evolving Cell Structure (ECS), is proposed. ECS is advantageous in that it not only captures adequate statistical information for outlier detection but also can be updated incrementally and efficiently. As for the future research work, we plan to incorporate time concept into ECS to reflect the effect of time on the evolving structure. The performance of the proposed method will also be evaluated using synthetic and real-life high-dimensional streaming data sets.

References

[1] Charu C. Aggarwal and Philip S. Yu. 2005. An effective and efficient algorithm for high-dimensional outlier detection. *VLDB Journal*, 14: 211-221, Springer-Verlag Publisher.

[2] Daniel Barbara. 2002. Requirements for clustering data streams. *ACM SIGKDD Explorations Newsletter*, Volume 3, Issue 2, 23 - 27, ACM Press.

[3] Ji Zhang and Hai Wang. 2006. Detecting Outlying Subspaces for High-dimensional Data: the New Task, Algorithms and Performance. *Knowledge and Information Systems (KAIS)*, Springer-Verlag Publisher.